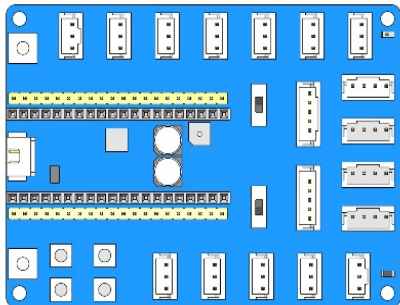




Lesson 4- Testing LDR on Pico Expansion Board

In this lesson, we will use Raspberry Pi Pico to testing LDR function on Pico Expansion Board ,control the LED and buzzer working by the LDR

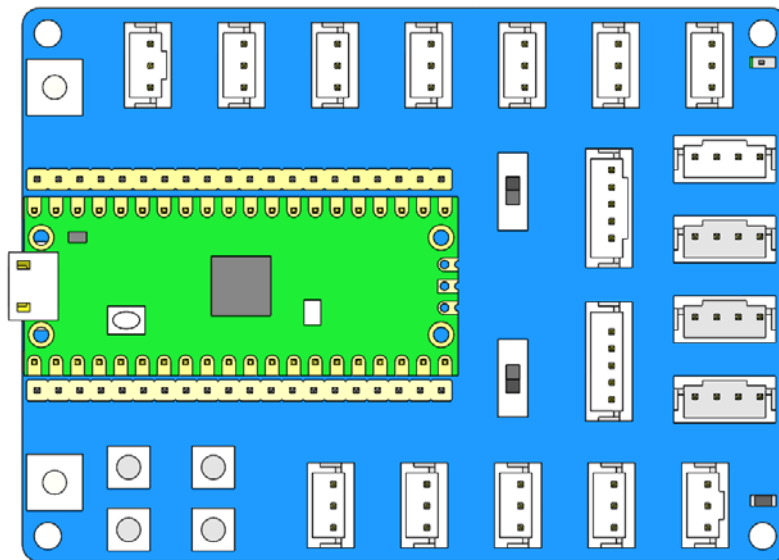
1.Component List

Component	QTY	Picture
Pico Expansion Board	1	
Raspberry Pi Pico	1	
USB Cable	1	

2. Circuit

Put assemble the Raspberry Pico on the Pico Expansion board

Hardware connection:



3. Upload Arduino code to Pico and testing

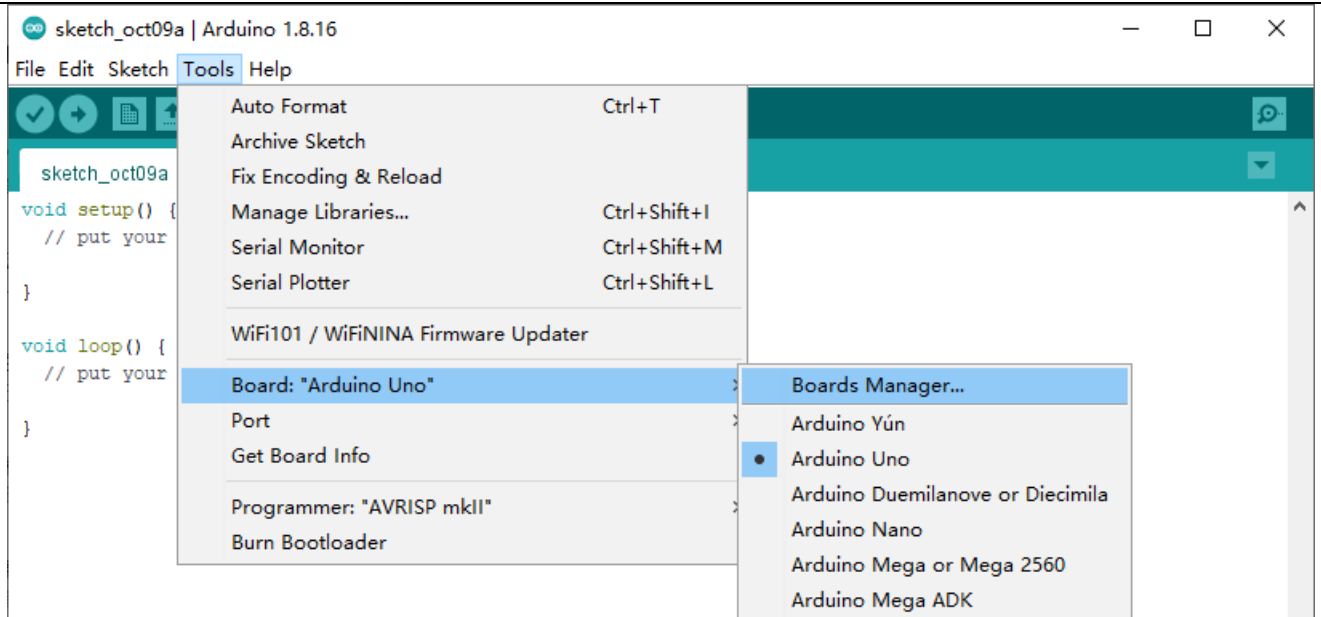
Configuring the Operating Environment For Arduino

Please refer to the document “[3 Configuring the Operating Environment For Arduino](#)” which saved in ‘[Pico_Extension_Board Kit Tutorial](#)’

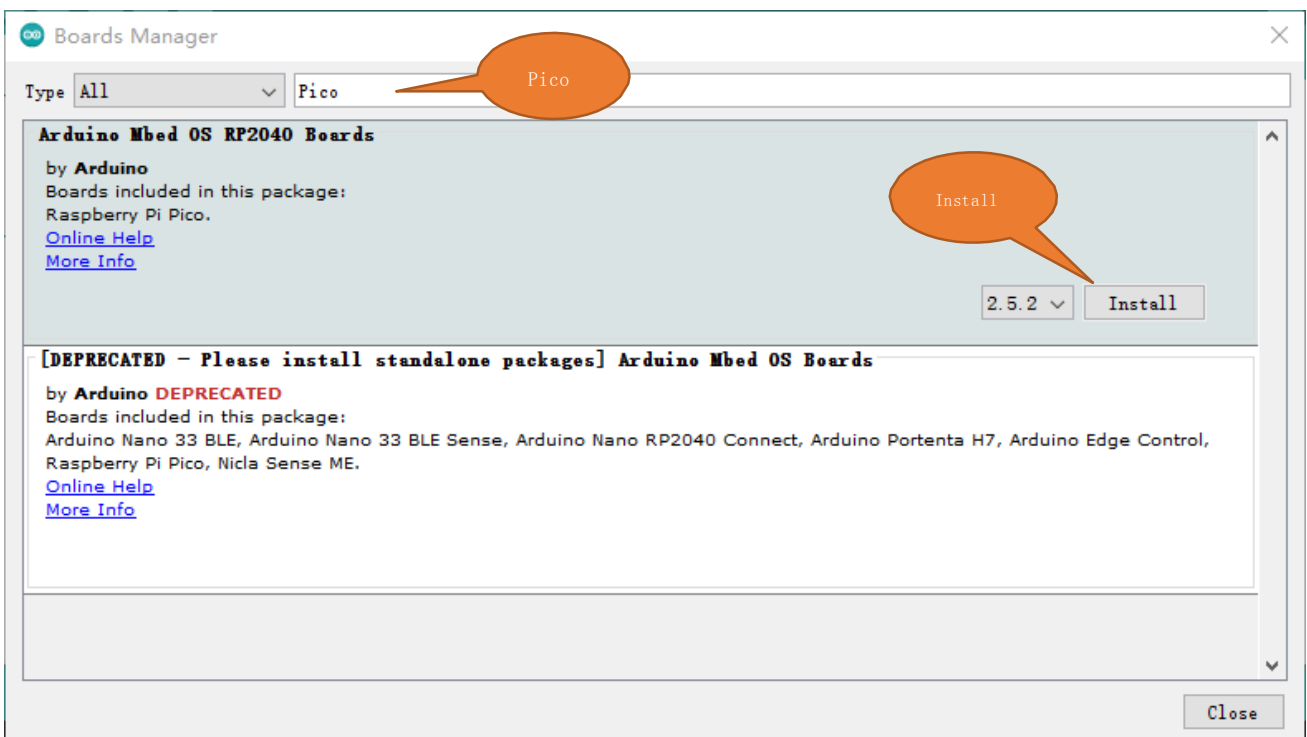
3.1 Installation of Development Board Support Package for Pico

3.1.1 Make sure your network is of good connection.

3.1.2 Open Arduino IDE. Click Tools>**Board**>**Boards Manager...**on the menu bar.



3.1.3 Enter Pico in the searching box, select "Arduino Mbed OS RP2040 Boards" and click on Install.



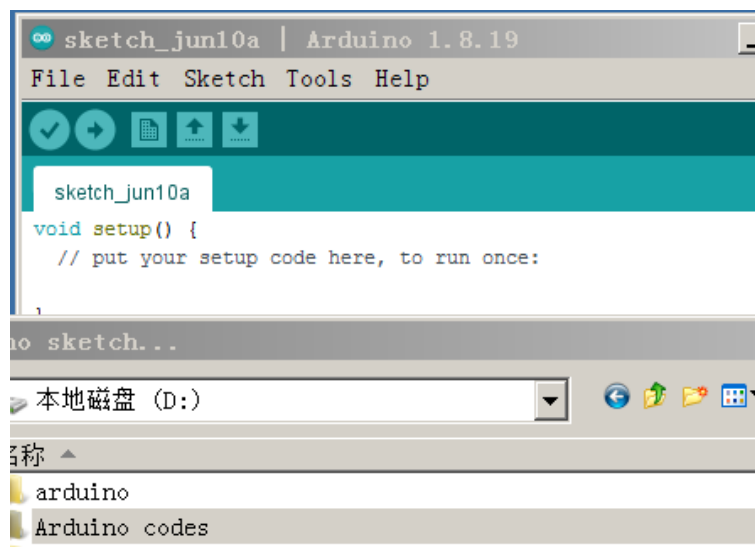
3.1.4 Click Yes in the pop-up "dpinst-amd64.exe" installation window. (Without it, you will fail to communicate with Arduino.) Thus far, we have finished installing the development support package.

3.2 Uploading testing Arduino code to Pico

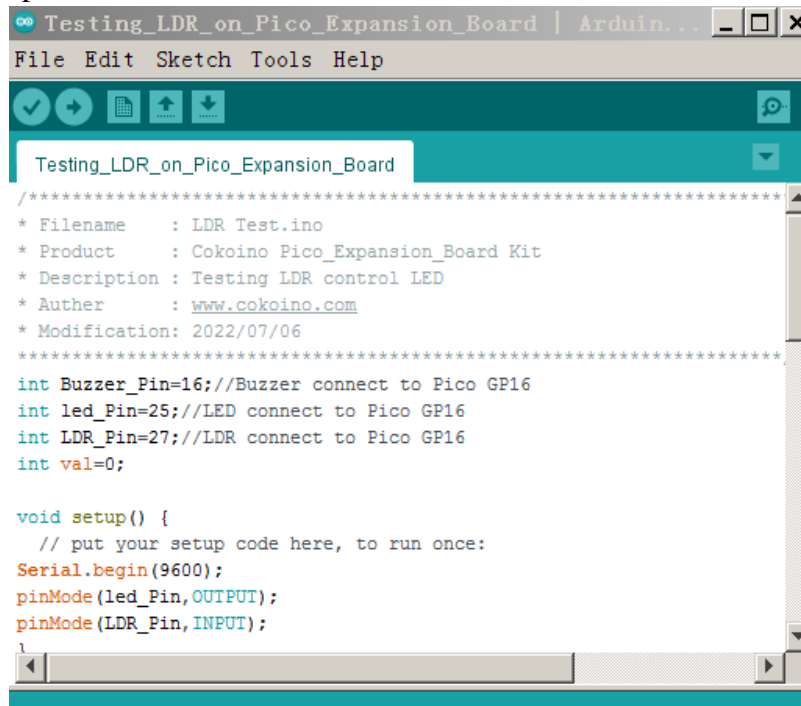
Codes used in this lesson are saved in "[Pico_Extension_Board Kit Tutorial\Codes\Arduino_Codes](#)". You can move the codes to any location. For example, we save the codes in Disk(D) with the path of "[D:/ Arduino codes](#)".

Testing LDR on Pico Expansion Board

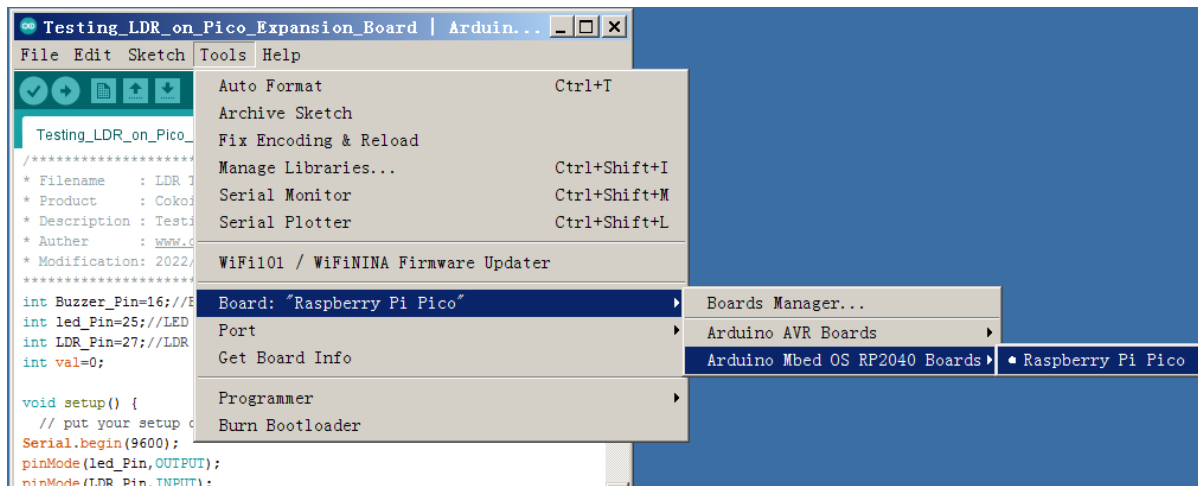
Open "[Arduino](#)" IDE, click "[File](#)"--"[Open](#)"--"[D:](#)"--"[Arduino codes](#)".



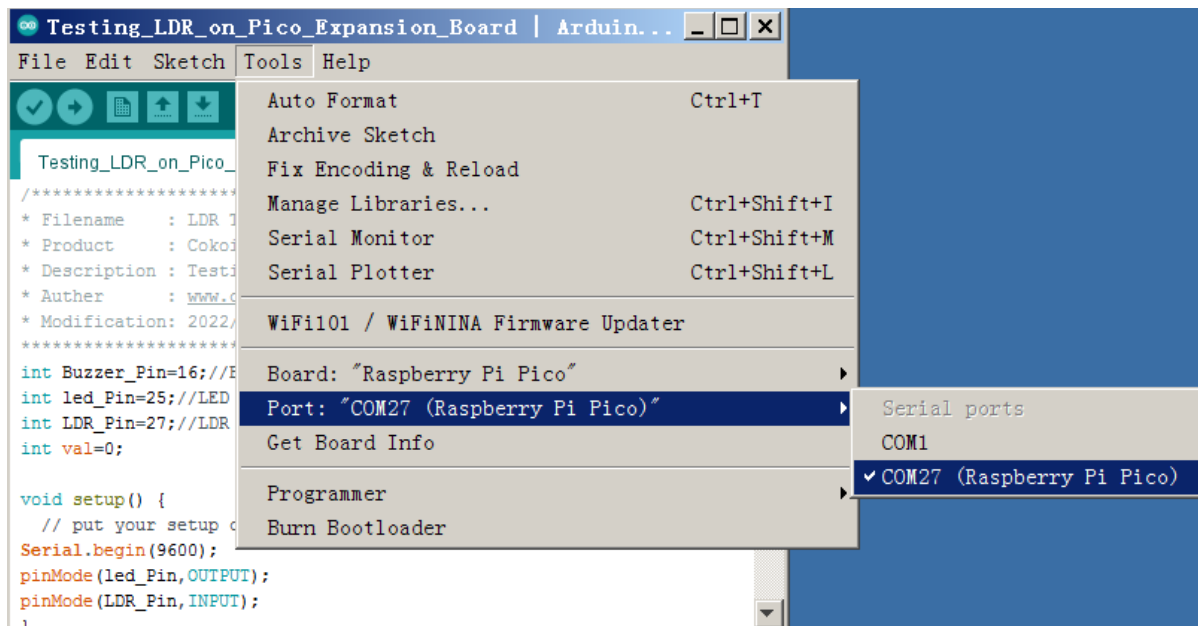
Expand folder "[Testing LDR on Pico Expansion Board](#)" and double click "[Testing LDR on Pico Expansion Board](#)" open it. As shown in the illustration below.



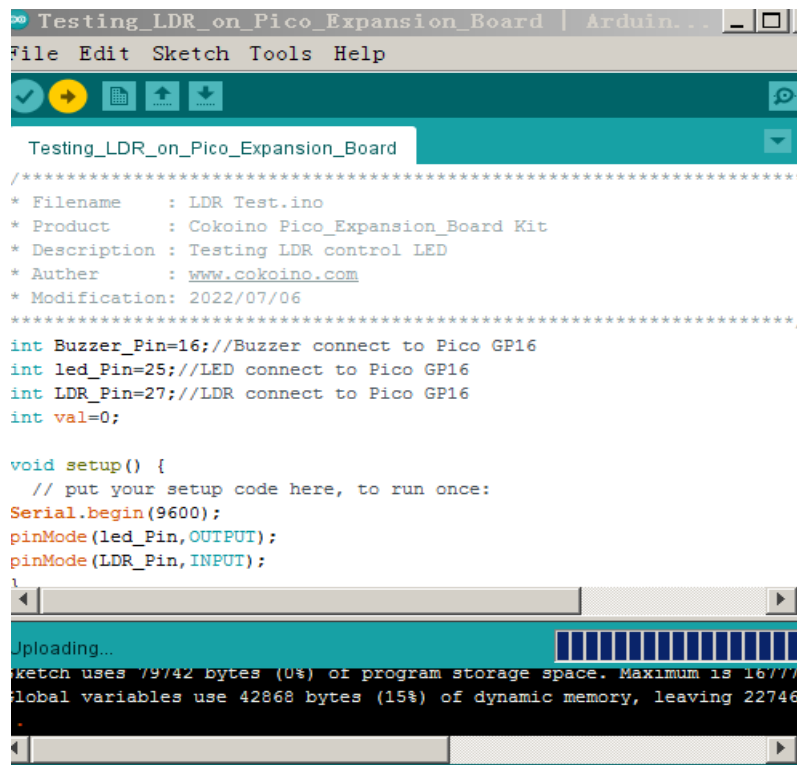
Make sure Raspberry Pi Pico has been connected with the computer by usb cable. Then click “Tools”-“Board”-“Arduino Mbed OS RP2040 Boards”-“Raspberry Pi Pico” As shown in the illustration below.



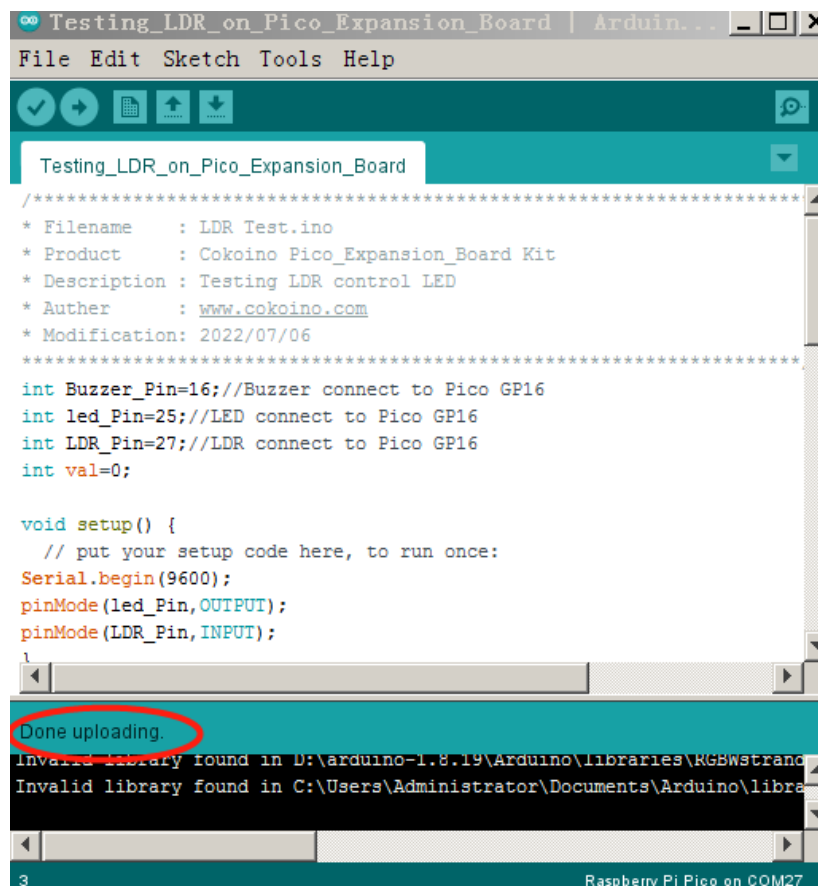
Click “Tools”-“Port” to select the Serial ports for Raspberry Pi Pico .As shown in the illustration below.



Upload the code to Raspberry Pi Pico as shown in the illustration below.



Upload succeeded as shown in the illustration below.



Notes:

- 1.Be sure to keep pressing the “BOOTSEL” button before powering the Pico, otherwise the firmware will not download successfully at the first time.
- 2.At the first time you use Arduino to upload sketch for Pico, you don't need to select port. After that, each time before uploading sketch, please check whether the port has been selected; otherwise, the downloading may fail.
- 3.Sometimes when using, Pico may lose firmware due to the code and fail to work. At this point, you can upload firmware for Pico as mentioned above

4. Testing results

After upload the testing code to RaspberryPico, please re-plug the usb cable in the Pico.

Place the Pico expansion board in a light environment , you will see the LED light on the Pico and hear the beep of the expansion board buzzer. Cover the LDR on the expansion board, the LED on the Pico turn off, and the buzzer on the expansion board stop working .

Code as below:

```
int Buzzer_Pin=16;//Buzzer connect to Pico GP16
int led_Pin=25;//LED connect to Pico GP16
int LDR_Pin=27;//LDR connect to Pico GP16
int val=0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(led_Pin,OUTPUT);
  pinMode(LDR_Pin,INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int LDR_Value=analogRead(27);//Read value of LDR sensor on the Pico Extension Board
  Serial.println(LDR_Value);//print the value on the serial monitor,value(min)=0,value(max)=1024
  if(LDR_Value>=200)//Set instruction condition values,you can set it between 0~1024 as you want
  {
    Buzz_Test();// Buzz working
    LED_Test();//LED on
  }
  else
  {
    digitalWrite(led_Pin,LOW);//led off
  }
  delay(1000);
}

void Buzz_Test()
{
  for (char i = 0; i < 80; i++)
  {
    digitalWrite(Buzzer_Pin,HIGH);
    delay(2);
    digitalWrite(Buzzer_Pin,LOW);
    delay(2);
  }
}
```

```
}  
for (char i = 0; i < 100; i++)  
{  
    digitalWrite(Buzzer_Pin,HIGH);  
    delay(1);  
    digitalWrite(Buzzer_Pin,LOW);  
    delay(1);  
}  
}  
void LED_Test()  
{  
    digitalWrite(led_Pin,HIGH);  
}
```

What's Next?

THANK YOU for participating in this learning experience!

If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.

<http://cokoino.com/>

Thank you again for choosing Cokoino products.